REMARKS/ARGUMENTS

Claims 1, 2, 5-8, 10-12, 14-22, 24-28, 31-37, and 39 remain for consideration by the Examiner. Claim 1 is amended to include the limitation that the service connector interface has no knowledge of the referenced service prior to gaining the reference to the service. No new matter is added by the amendment.

Rejections Under 35 U.S.C. § 103

In the Office Action of September 15, 2003, the rejection was maintained for claims 1, 2, 5-8, 10-12, 14-22, 24-28, 31-36, and 39 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,366,916 ("Baer") in view of U.K. Patent No. 2,347,766 A ("Wilson"). The rejection of the claims based on the combination of these references is respectfully traversed based on the following remarks.

With reference to Applicant's specification, it may be useful to first discuss generally Applicant's invention. Briefly, the "present invention provides a mechanism for gaining access to the plug-ins" (see page 17, lines 4-5 with reference to Figure 2). The invention, according to the paragraph beginning at line 18 of page 17, addresses the situation where an application program 302 running in a local environment 308 has need of utilizing a service in the local or distributed environment 310. The invention comprises a service connector 304 that provides a "mechanism through which the application 302 (e.g. the client) can obtain a reference to the service 306 ... and use it." To this end, "the fundamental method utilized is denominated 'getService' and invoking it returns a reference to the actual service 306." In the paragraph at line 5, page 18, "Upon start-up, the application 302 first reads a configuration file...This provides an indicator to the application 302 such that ... the application 302 can instantiate the service connector 304." When the getService(version) method is called, the service connector 304 performs a lookup of the service instance, e.g., a specified version of the service desired by the application 302. The method of the invention addresses the problem of using services not known prior to runtime, and this technique is different than prior art systems that use directories and meta-directories to catalog already known information regarding available services (see, for example, Applicant's Background at page 5, lines 1-13). In varying language, each of the independent claims calls for gaining a

reference to a service and returning it to another service or client application and in many cases, claim invoking a service connector interface.

Baer teaches a different system and methods, which may best be understood by looking overall at the workings of Baer rather than trying to interpret isolated citations in Baer. The operation of the overall Baer architecture is explained in detail with reference to Figure 7 beginning at col. 19, line 27. The client application contains a master service adapter 703 and a logon client adapter 702 that are embedded in the client application and when "creating a new Client Adapter 705, for example, interfaces for Client Adapter 705 and Schema controller 707 must be defined". The logon client adapter 702 is used to establish the connection between the client application and the asset manager server (AMS). "The Client Application then sends a logon message to the Logon Client Adapter which turn constructs the Master Server Adapter 703..." "Next, the Client Application sends a getSessionTypes method to the Master Client Adapter 701..." which results in the master server 704 returning "a list of Client Adapters 705 to the Master Client Adapter 701 and then on to the Client Application." The Client Application sends a newSession message to the Master Server Adapter 703 "providing the unique ID of the desired Client Adapter 705", which results in the adapter 701 registering the "New Client Adapter 705."

Hence, the client application of Baer selects the client adapter 705 from a list provided by the AMS and then can use the client adapter's functions to access the assets or data in the data store 716. The "list" is apparently built and retained by the master server 704 and includes building a "directory of all Client Adapters 705 in the system, thereby registering the Client Adapters 705" (see col. 7, lines 20-25). Baer teaches "dynamic" identification and use of a new client adapter by a client application in the above sense, but the new client adapter is known to the AMS and included in its directory whose contents are noted in the list provided to the client application. Hence, Baer does not teach obtaining a reference of unknown but available services at runtime and then providing a reference to the client application as called for in the pending independent claims. The claimed invention does not require the user to select from a list of adapters and further respond with the ID of such adapter prior to using the functions of the adapter. As stated at page 5, lines 24-29 of Applicant's specification, the "system and method of the present invention advantageously provides a generic mechanism for a service to dynamically gain a reference to another service without requiring knowledge

of how to find the service or a particular version of the service on a distributed computer network."

Turning specifically to the Office Action, the Response to Arguments states that Baer teaches the client application making dynamic calls to the AMS. As noted above, Baer does teach that the client application communicates with the AMS to obtain a "new client adapter" for use in retrieving assets/data, and this communication is performed at runtime. However, Baer teaches maintaining a directory of all client adapters in the system, and then at runtime providing the list of the client adapters to the client application, which must then respond by choosing an adapter by its unique ID. In contrast, the claims generally call for instantiating the service connector interface at the second service or client application and then gaining reference to the first service for use by the second service. The first service is unknown to the service connector prior to the gaining as noted in the amendment to claim 1. The Response to Arguments also indicates that a new instance of the "IDLSessionInfol interface" is "a reference used by the Client Application for asset retrieval." However, a reference according to the invention is information useful by a service for using, running, calling, and the like a second service and not an interface used for retrieving data (although the second service may provide such functions).

More particularly, with reference to the pending claim language, Claim 1 is directed to a method calling for "enabling definition of a service connector interface in conjunction with said first service" (i.e., the service being accessed) and then "subsequently invoking said service connector interface in conjunction with said second service...including instantiating said service connector interface at said second service." The method calls for an application at the second service to perform the invoking of the service connector interface and does so by "reading a configuration file providing an indicator for said service connector interface." Claim 1 requires that a service connector interface be defined for the service being accessed and that a configuration file be made available to an application that then instantiates the service connector based on the configuration file. Further, Claim 1 calls for gaining reference to the first service by the second service by "retrieving a service instance at said service connector interface," "obtaining a service reference from said first service," and "returning said service reference obtained from said first service to said second service." Because each of these limitations is not shown or even suggested by Baer and Wilson, Claim 1 is believed

allowable over the cited references.

Baer teaches using a client application layer 20 to allow a user to interact with an asset manager system (AMS) 30, which provides an interface with the "assets" in a data store 40 (see Baer at col. 3, lines 49-67). An "asset" is defined "to be a set of related data, or meta data, for a document." Baer provides no teaching of a second service gaining a reference to a first service in the manner as called for in Claim 1. Baer does not teach that the client adapters are unknown to the AMS 30 but instead teaches that the AMS 30 maintains a directory for use in providing a list to the client application). This is significant because the Office Action cites Baer as teaching a first service with the AMS 30 and a second service (and application program) with the client application level 20. Further, it does not appear that Baer discusses unknown services, i.e., unknown to the AMS 30, that are being accessed by the client application but rather teaches a data store interface or connection method, which is different from the method of Claim 1.

Specifically, Claim 1 calls for "gaining reference to said first service by said second service" and then provides the specific steps of retrieving, obtaining, and returning that are not shown in Baer. The Office Action cites Baer at col. 8, lines 1-19 for teaching the gaining element of Claim 1 but at this citation, Baer explains how a Master Client Adapter 701 acts on a request from a Client Application to retrieve an asset from the data store 40. Baer does not suggest that a reference to a service should be retrieved and clearly does not teach retrieving an instance at a service connector interface, obtaining a service reference from the first service, and then returning the service reference to the second service. Further, the gaining step is carried out by "an application program operative in conjunction with said second service." This feature is not shown by Baer as suggested by the Office Action with the "Client Application" which instead teaches using the AMC 200 to manipulate a middleware application AMS 30 to access data (not a set of unknown services that are unknown by the client application and may have differing interface requirements).

The instantiating called for in Claim 1 is required to be performed by reading a configuration file providing an indicator for said service connector interface." The Office Action states that Baer is silent as to such instantiating but cites Wilson for teaching a Configuration File at page 3, lines 4-29. However, Wilson is directed toward a method of printing from the Internet using a web browser. A configuration file is used for configuring a

printer, but there is no motivation in Baer to modify its teaching to use a configuration file (let alone a printer configuration file) during the instantiation of its connector adapters. The Response to Arguments indicates that Wilson provides motivation to modify Baer, but Applicant disagrees as Baer is addressing how to access data which may have a number of formats but does not generally discuss how to obtain additional, unknown services in a distributed computing environment. Hence, Claim 1 is believed allowable because this additional limitation of Claim 1 is not shown by Baer and Baer's deficiencies are not overcome by Wilson.

Claims 2 and 5-8 depend from Claim 1 and are believed allowable for at least the reasons for allowing Claim 1. Further, Claims 6 and 8 are directed to the concept of defaulting to instantiating and then referencing a latest version or latest instance of a service being accessed or requested by client application or "second service." Wilson is cited for teaching the retrieval of a particular version. However, again, Baer does not teach obtaining a reference to and then access to a second service. Hence, there is no discussion of versions of the data nor the need (i.e., lack of motivation found in Baer) for modifying or improving the Baer process to include the version features of Claims 6 and 8, and these claims are believed allowable for this additional reason. Baer does not even teach that the numerous versions of the client adapters may be made available to the client application in its lists or directories.

Independent Claim 10 is directed to a computer program product with some limitations similar to Claim 1 but further calls for gaining reference to the first service by the second service to include retrieving an instance of the first service at the service connector interface, obtaining a service reference from the first service, and then returning said reference to the second service. Further, Claim 10 as amended requires that the first service be in "a distributed environment" and that the second service be in "a local environment" and that "the service connector interface encapsulating logic necessary to retrieve service instances in the distributed environment not in a directory service in the local environment. Claim 10 is believed allowable for the reasons for allowing Claim 1 and also because Baer fails to teach obtaining generic services within from a distributed environment. Hence, Baer fails to teach that the first and second services are in different computing environments and that the service connector interface that is invoked within the local environment of the second service includes embedded or hidden logic necessary to obtain a reference to the first service

in the distributed environment (i.e., the logic does not have to be embedded in an application in the local environment or in objects in memory at the second service). Independent Claim 10, and Claims 11, 12, and 14-18 that depend from Claim 10, are believed nonobvious in light of the teachings of Baer when combined with Wilson.

Independent Claim 19 is directed to a method similar to Claim 1 and is believed allowable for at least the reasons for allowing Claim 1 but further includes the limitation that a particular version of the first service can be specified by the second service. As discussed with reference to Claims 6 and 8, Baer fails to teach this added limitation. There is no motivation to combine Wilson with the teachings of Baer, and further, there is no teaching in Wilson that the "version" to be considered is of a service. Claim 19, and Claims 20-22 and 24-27 that depend from Claim 19, are not taught or suggested by Baer in view of Wilson and are believed in condition for allowance. Additionally, no mention of the "version" limitation is provided in the Response to Arguments portion of the Office Action.

Independent Claim 28 is directed to a system for providing dynamic references between services and is believed allowable for the reasons for allowing Claim 1 (as amended to include means for instantiating a service connector at the second service) but also because it calls for means for enabling definition of a service connector interface in conjunction with the first service that includes means for developing a computer program module adhering to the service connector interface in conjunction with the first service. Claim 28 is not taught or suggested by Baer alone or in combination with Wilson and Claim 28, and Claims 30-32 and 34-36 that depend there from, are allowable over the art of record.

Independent Claim 37 is directed to a core profile engine as shown in Figure 2 with a pluggable interface for use in attaching to service modules and including a service connector associated with each attached service module. Claim 37 calls for "a pluggable interface attaching to the plug-in service modules" "wherein the pluggable interface further includes a service connector...adapted to receive the service request from the application programming interface and to return a reference to the one service module ..." Further, as amended, Claim 37 calls for the plug-in service modules to be selected from an authorization plug-in, an authentication plug-in, a notification plug-in, a log plug-in, a group plug-in, an entity identification factory plug-in, and a replication plug-in.

The Office Action states that Baer teaches all of the limitations of Claim 37, but as

discussed with reference to Claim 1, Baer fails to teach "a service connector associated with each of the attached plug-in service modules that is adapted to receive the service request from the application programming interface and to return a reference to the one service module providing the service based on the storage location." Baer fails to teach the idea of receiving a service request and responding by discovering a reference to the service and returning the reference to the requesting service. Further, Baer does not teach performing such a step based on a storage location. Additionally, the amended Claim 37 calls for specific set of service plug-ins that are not taught or suggested by Baer (or Wilson). The Office Action points to a "Plug-in Factory 708 and Plug-in Container 709 in Baer, but Baer at col. 5, line 65 states that these correspond to Plug-ins module 312 of Figure 4. The resources module 302 includes these plug-ins to allow a client adapter module 300 to "accomplish tasks required by the Client Application 20" (see col. 5, lines 19-21) but no discussion is provided on how this module operates or what the plug-ins may be. Without further explanation, Baer does not make the engine of Claim 37 obvious and does not anticipate the specific plug-ins called for in Claim 37.

Independent Claim 39 is directed to a method of providing an application with a reference to or access to a particular service. Baer in Figure 4 indicates that plug-ins 312 may be available through the AMS 30 to a client application 20. However, there is no teaching of reading a configuration file to determine an indicator to a service or of instantiating with the application a service connector for service based on the indicator. Baer instead is directed toward facilitating querying a data store 40 to retrieve data or "assets." Further, Baer does not teach operating an application to request identification of an interface implemented by the referenced service and then operating the service connector to retrieve and return the interface identification to the application for use in utilizing the reference service. Baer only mentions the "External Services" in passing as interfacing with these services is not the thrust of the Baer invention. There is no indication that the services 322 vary over time (requiring access to new versions) or that the plug-ins 312 are not known and must be discovered (gained reference to). Because Baer was addressing a different problem (e.g., how to interface with data in a database), Baer fails to teach or suggest each and every element of the method of Claim 39. Hence, Claim 39 is believed in condition for allowance.

Rejections Under 35 U.S.C. § 103 of Claim 37

In the Office Action, claim 37 was rejected under 35 U.S.C. §103(a) as being unpatentable over Baer in view of U.S. Patent No. 5,872,966 ("Burg") further in view of published U.S. Patent No. 2003/0120597 ("Drummond") and further in view of U.S. Patent No. 6,553,413 ("Leighton"). The rejection of the claim 37 based on the combination of these references is respectfully traversed based on the following remarks.

Initially, Applicant respectfully requests that the finality of the Office Action be withdrawn. In the last response, Applicant amended claim 37 only to include limitations previously presented in dependent claim 38, and so, no new subject matter was presented to the Office. However, the Office Action cites 3 new references against this claim, i.e., Burg, Drummond, and Leighton. As Applicant has not had an opportunity to review and respond to these references, the Office Action should have been non-final.

However, the combination of these references fails to teach or suggest all of the limitations of claim 37. Independent Claim 37 is directed to a core profile engine as shown in Figure 2 with a pluggable interface for use in attaching to service modules and including a service connector associated with each attached service module. Claim 37 calls for "a pluggable interface attaching to the plug-in service modules" "wherein the pluggable interface further includes a service connector...adapted to receive the service request from the application programming interface and to return a reference to the one service module ..." Further, Claim 37 calls for the plug-in service modules to be selected from an authorization plug-in, an authentication plug-in, a notification plug-in, a log plug-in, a group plug-in, an entity identification factory plug-in, and a replication plug-in.

The Office Action states that Baer teaches all of the limitations of Claim 37 except the specific plug-ins. However, as discussed with reference to Claim 1, Baer fails to teach "a service connector associated with each of the attached plug-in service modules that is adapted to receive the service request from the application programming interface and to return a reference to the one service module providing the service based on the storage location." Baer fails to teach the idea of receiving a service request and responding by discovering a reference to the service and returning the reference to the requesting service. Further, Baer does not teach performing such a step based on a storage location. Hence, Baer does not teach or suggest each of the limitations of claim 37 except for the specific plug-ins, and claim

37 is believed in condition for allowance.

Additionally, the amended Claim 37 calls for specific set of service plug-ins that are not taught or suggested by Baer as is noted in the Office Action. The Office Action cites to Burg, Drummond, and Leighton to, in combination, teach each of the plug-ins of claim 37. However, none of these references provide the missing teaching of Baer with regard to the application programming interface, and therefore, the combination of references fails to teach the engine of claim 37. The invention of claim 37 is not the specific plug-ins utilized but the combination of the plug-ins, for which references are obtained, along with the uniquely adapted application programming interface, and the 3 new references are not cited in the Office Action for teaching the application programming interface or for obtaining references to the plug-ins.

Conclusions

The pending claims are believed allowable over these additional references taken alone or in combination with the other cited references.

No fee is believed due with this response, but any fee deficiency associated with this submittal may be charged to Deposit Account No. 50-1123.

It is respectfully requested that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

Date 11/14/03

Kent Lembke, Reg. No. 44,866
Hogan & Hartson LLP
One Tabor Center
1200 17th Street, Suite 1500
Denver, Colorado 80202
Telephone: (720) 406-5378
Fax: (720) 406-5301